Listing of Claims:

1. Please Cancel

2. (Amended) A processor for performing a multiply-add instruction on a multiplicand A, a multiplier B, and an addend C, to calculate a result D, each of A, B and C being a double-precision floating point number, the result D being a canonical-form extended-precision floating point number having a high order component and a low order component, each double-precision number and each of the high and low order components of an extended-precision number comprising an exponent and a mantissa,

the processor operating on clock cycles and comprising

a multiplier, an adder, and a normalizer for computing intermediate results in the computation of the multiply-add instruction,

a rounder for rounding intermediate results to the result D,

a data path in the processor to permit data to flow in sequence from the multiplier to the adder to the normalizer to the rounder, and

a set of result registers accepting output from the rounder, for sequentially storing the mantissa of each of the high and low order components of D,

the post-adder data path, the normalizer and the rounder each having a data width sufficient to represent post-adder intermediate results whereby both of the high and low order components of the correctly-rounded result D may be computed, and

the data path, the multiplier, the adder, the normalizer and the rounder being arranged to permit the respective mantissas of the high order component of D and of the low order component of D to be stored to the set of result registers on sequential clock cycles of the processor, and where the

~~The~~ processor ~~of claim 1 comprising~~ logic control determines ~~to determine~~ differential values of the exponents of A, B and C and to carry out operations in the processor, the logic control providing that

a. where the exponent of C is greater than the sum of the exponents of A and B by a predetermined limit,

   the mantissa of the high order component of result D is computed by taking the mantissa of C, and

   the mantissa of the low order component of result D is computed by taking the normalized, rounded result of the multiplier having as inputs the mantissas of A and B,

b. where the exponent of C is within the predetermined limit of the sum of the exponents of A and B,

   the mantissa of the high order component of result D is computed by taking the normalized, rounded result of the multiplier and the adder having as inputs the mantissas of A, B and C, and

   the mantissa of the low order component of result D is computed by wrapping the low order bits of the result of the normalizer to a first input to the adder, and by supplying a selected one of two predetermined values to a second input to the adder, the selection of the predetermined values being made based on the rounding of the high order component of result D, whereby the low order component of result D is decremented where the high order component is incremented by the rounder,

c. where the exponent of C is less than the sum of the exponents of A and B by the predetermined limit,

   the mantissa of the high order component of result D is computed by taking the normalized, rounded result of the multiplier and the adder having as inputs the mantissas of A, B and C, and

   the mantissa of the low order component of result D is computed by wrapping a negative value of high order component of result D to the processor as an addend, and resupplying A and B as multiplicand and multiplier to calculate a remainder value and by further wrapping the remainder and C to the processor to obtain the computed sum of the

remainder value and C.

3. (Original) The processor of claim 2 in which the predetermined limit is equal to the length of the mantissas of the double-precision and the extended-precision numbers

4. (Original) The processor of claim 3 in which the length of the mantissas of the double-precision and the extended-precision numbers is 53 bits and in which the predetermined limit is 53.

5. (Original) A method for computing the mantissa of a canonical form extended-precision number for the result D for the multiply-add instruction A * B + C,

where A, B and C are double-precision numbers, the result D being a canonical-form extended-precision floating point number having a high order component and a low order component, each double-precision number and each of the high and low order components of an extended-precision number comprising an exponent and a mantissa,

the method implemented on a computer processor, the processor comprising

an alignment shifter, a multiplier, an adder, an incrementer, and a normalizer for computing intermediate results in the computation of the multiply-add instruction,

a rounder for rounding intermediate results to the result D,

a data path in the processor to permit data to flow in sequence from the multiplier to the adder to the normalizer to the rounder, and

a set of result registers connected to the rounder, for sequentially storing the mantissa of each of the high and low order components of D,

the method comprising the steps of:

a. shifting the mantissa of C using the alignment shifter, to form a shifted mantissa of C, having a low order portion and a high order portion, the shifting being based on the relative values of the exponents of A, B and C
b. computing partial products of the mantissas of A and B,

c. compressing the low order portion of the shifted mantissa of C with the partial products,

d. adding the compressed low order portion of the shifted mantissa of C and the partial products, using the adder, to generate a carry bit and an add-out value, the add-out value having a binary representation with sufficient bits to represent the added compressed low order portion of the shifted mantissa of C and the partial products,

e. conditionally incrementing the high order portion of the shifted mantissa of C using the incrementer, the increment being based on the carry bit of the adder for the addition of the compressed low order portion of the shifted mantissa of C and the partial products,

f. concatenating the high order word of the shifted mantissa of C with the add-out value, the concatenated binary value representing a pair of words comprising a dh value representing the high order word of the addition and multiplication result and a pre-dl value representing a preliminary value for the low order word of the addition and multiplication result,

g. normalizing and rounding the concatenated binary value using the normalizer and rounder,

h. providing the normalized and rounded dh value to a selected one of the set of result registers on a high order result clock cycle,

i. determining if the normalized pre-dl value requires modification, modifying the normalized pre-dl value where required, generating a rounded dl value using the normalized pre-dl value, where the dl value is the low order word of the result D, and

j. providing the dl value to a selected one of the set of result registers on a low order clock cycle different from the high order result clock cycle.

6. (Amended) The <u>processor</u> ~~method~~ of claim 3 in which the step of determining if the pre-dl requires modification comprises the step of comparing the relative values of the exponents of A,

B and C.

7. (Amended) The <u>processor</u> ~~method~~ of claim 6 in which the comparison of relative values of the exponents of A, B and C comprises the step of determining whether the exponent of C is greater than, less than, or within a predetermined limit of the sum of the exponents of A and B.

8. (Amended) The <u>processor</u> ~~method~~ of claim 7 in which the predetermined limit is equal to the length of the mantissas of the double-precision and the extended-precision numbers.

9. (Amended) The <u>processor</u> ~~method~~ of claim 8 in which the length of the mantissas of the double-precision and the extended-precision numbers is 53 bits and in which the predetermined limit is 53.

10. (Amended) The <u>processor</u> ~~method~~ of claim 3 in which the step of determining if the normalized pre-dl value requires modification comprises the step of comparing the relative values of the exponents of A, B and C and

a. where the exponent of C is greater than the sum of the exponents of A and B by a predetermined limit, determining that the normalized pre-dl does not require modification,
b. where the exponent of C is within a predetermined limit of the sum of the exponents of A and B,

    determining that the normalized pre-dl is to be potentially modified by wrapping the normalized pre-dl value to a first input to the adder, and by supplying a selected one of two predetermined values to a second input to the adder, the selection of the predetermined values being made based on the rounding of dh, whereby the normalized pre-dl is decremented where dh is incremented by the rounder,

c. where the exponent of C is less than the sum of the exponents of A and B by a predetermined limit,

the normalized pre-dl is determined to require modification, the modification to be executed by wrapping a negative value of dh to the processor as the addend C, and by inputting the initial A and B values as multiplicand and multiplier to calculate a remainder value equal to A*B – dh, and by further wrapping the remainder and inputting the initial C value to the processor to modify the normalized pre-dl to be the computed sum of the remainder value and C.

11. Please Cancel

12. (Amended) In a fused mutliply-add processor, an improvement for outputting the mantissa of a canonical form extended-precision number for the result D for the multiply-add instruction A * B + C,

where A, B and C are double-precision numbers, the result D being a canonical-form extended-precision floating point number having a high order component and a low order component, each double-precision number and each of the high and low order components of an extended-precision number comprising an exponent and a mantissa,

the improvement being characterized by,

the post-adder components in the improved fused multiply-add processor for computing intermediate result numbers comprising a post-adder data path, a normalizer and a rounder, and associated registers, each having a bit-width sufficient to represent the mantissas of the intermediate result numbers so as to permit the computation of the mantissa of the extended-precision result D, and logic control in the improved fused multiply-add processor to provide a high order word mantissa and a low order word mantissa of the extended-precision result D to a set of result registers in separate clock cycles, and where the

~~The improved~~ fused multiply-add processor ~~of claim 11 in which the~~ logic control further provides a wrap back of a low order portion of an intermediate result value to the adder and

further supplies a selected predetermined value to the adder to decrement the low order word mantissa of the extended-precision result where the high order word mantissa of the extended-precision result is incremented by the rounder.

13. (Amended) The improved fused multiply-add processor of claim <u>12</u> ~~11~~ in which the logic control further provides a wrap back of intermediate values to the input of the fused multiply-add processor to provide for adjustment to the low order word mantissa of the extended-precision result, where the value of the addend for execution of the multiply-add instruction is truncated during execution of the instruction.

14. (Original) An improved fused multiply-add processor for computing the mantissa of a canonical form extended-precision number for the result D for the multiply-add instruction A * B + C,

> where A, B and C are double-precision numbers, the result D being a canonical-form extended-precision floating point number having a high order component and a low order component, each double-precision number and each of the high and low order components of an extended-precision number comprising an exponent and a mantissa,

the fused multiply-add processor comprising

> an alignment shifter, a multiplier, an adder, an incrementer, and a normalizer for computing intermediate results in the computation of the multiply-add instruction,
>
> a rounder for rounding intermediate results,
>
> a data path in the processor to permit data to flow in sequence from the multiplier to the adder to the normalizer to the rounder, and
>
> a set of result registers taking the results of the rounder as input,

the improvement being characterized by

a. the post-adder data path, the normalizer and the rounder each having a data width sufficient to represent post-adder intermediate results whereby both of the high and low order components of the correctly-rounded result D may be computed,

b. logic control for the improved fused multiply-add processor to carry out the following steps:

i. shifting the mantissa of C using the alignment shifter, to form a shifted mantissa of C, having a low order portion and a high order portion, the shifting being based on the relative values of the exponents of A, B and C

ii. computing partial products of the mantissas of A and B,

iii. compressing the low order portion of the shifted mantissa of C with the partial products,

iv. adding the compressed low order portion of the shifted mantissa of C and the partial products, using the adder, to generate a carry bit and an add- out value, the add-out value having a binary representation with sufficient bits to represent the added compressed low order portion of the shifted mantissa of C and the partial products,

v. conditionally incrementing the high order portion of the shifted mantissa of C using the incrementer, the increment being based on the carry bit of the adder for the addition of the compressed low order portion of the shifted mantissa of C and the partial products,

vi. concatenating the high order word of the shifted mantissa of C with the add-out value, the concatenated binary value representing a pair of words comprising a dh value representing the high order word of the addition and multiplication result and a pre-dl value representing a preliminary value for the low order word of the addition and multiplication result,

vii. normalizing and rounding the concatenated binary value using the normalizer and rounder,

viii. providing the normalized and rounded dh value to one of the set of result registers on a high order result clock cycle,

ix. determining if the normalized pre-dl value requires modification, modifying the normalized pre-dl value where required, generating a

rounded dl value using the normalized pre-dl value, where the dl value is the low order word of the result D, and

x. providing the dl value to a selected one of the set of result registers on a low order clock cycle different from the high order result clock cycle.

15. (Original) The improved fused multiply-add processor of claim 14 in which the control logic for carrying out the step of determining if the normalized pre-dl value requires modification comprises the step of comparing the relative values of the exponents of A, B and C and

a. where the exponent of C is greater than the sum of the exponents of A and B by a predetermined limit, determining that the normalized pre-dl does not require modification,

b. where the exponent of C is within a predetermined limit of the sum of the exponents of A and B,

   determining that the normalized pre-dl is to be potentially modified by wrapping the normalized pre-dl value to a first input to the adder, and by supplying a selected one of two predetermined values to a second input to the adder, the selection of the predetermined values being made based on the rounding of dh, whereby the normalized pre-dl is decremented where dh is incremented by the rounder,

c. where the exponent of C is less than the sum of the exponents of A and B by a predetermined limit,

   the normalized pre-dl is determined to require modification, the modification to be executed by wrapping a negative value of dh to the processor as the addend C, and by inputting the initial A and B values as multiplicand and multiplier to calculate a remainder value equal to A*B – dh, and by further wrapping the remainder and inputting the initial C value to the processor to modify the normalized pre-dl to be the computed sum of the remainder value and C.

16. The improved fused multiply-add processor of claim 15 in which the predetermined limit is equal to the length of the mantissas of the double-precision and the extended-precision numbers

17. The improved fused multiply-add processor of claim 15 in which the length of the mantissas of the double-precision and the extended-precision numbers is 53 bits and in which the predetermined limit is 53.